

**This Page Is Inserted by IFW Operations  
and is not a part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

1c971 U.S. PTO  
10/029966  
12/31/01



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원 번호 : 특허출원 2001년 제 31124 호  
Application Number PATENT-2001-0031124

출원 년 월 일 : 2001년 06월 04일  
Date of Application JUN 04, 2001

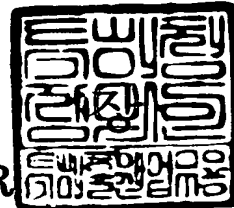
출원 인 : 삼성전자 주식회사  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2001 년 08 월 27 일

특 허 청

COMMISSIONER



	<b>【서지사항】</b>
<b>【서류명】</b>	특허출원서
<b>【권리구분】</b>	특허
<b>【수신처】</b>	특허청장
<b>【참조번호】</b>	0001
<b>【제출일자】</b>	2001.06.04
<b>【국제특허분류】</b>	G11C
<b>【발명의 명칭】</b>	플래시 메모리 관리방법
<b>【발명의 영문명칭】</b>	Flash memory management method
<b>【출원인】</b>	
<b>【명칭】</b>	삼성전자 주식회사
<b>【출원인코드】</b>	1-1998-104271-3
<b>【대리인】</b>	
<b>【성명】</b>	이영필
<b>【대리인코드】</b>	9-1998-000334-6
<b>【포괄위임등록번호】</b>	1999-009556-9
<b>【대리인】</b>	
<b>【성명】</b>	이해영
<b>【대리인코드】</b>	9-1999-000227-4
<b>【포괄위임등록번호】</b>	2000-002816-9
<b>【발명자】</b>	
<b>【성명의 국문표기】</b>	김범수
<b>【성명의 영문표기】</b>	KIM, Bum Soo
<b>【주민등록번호】</b>	690121-1019710
<b>【우편번호】</b>	431-070
<b>【주소】</b>	경기도 안양시 동안구 평촌동 초원마을 대림아파트 203동 1105호
<b>【국적】</b>	KR
<b>【발명자】</b>	
<b>【성명의 국문표기】</b>	조유근
<b>【성명의 영문표기】</b>	CHO, Yoo Kun
<b>【주민등록번호】</b>	490324-1090715
<b>【우편번호】</b>	135-240

【주소】	서울특별시 강남구 개포동 654 현대아파트 209동 401호
【국적】	KR
【발명자】	
【성명의 국문표기】	민상렬
【성명의 영문표기】	MIN,Sang Lyul
【주민등록번호】	610818-1029516
【우편번호】	137-060
【주소】	서울특별시 서초구 방배동 대우효령아파트 105동 906호
【국적】	KR
【발명자】	
【성명의 국문표기】	노삼혁
【성명의 영문표기】	NOH,Sam Hyuk
【주민등록번호】	630321-1052410
【우편번호】	121-160
【주소】	서울특별시 마포구 상수동 93-103 101호
【국적】	KR
【발명자】	
【성명의 국문표기】	이귀영
【성명의 영문표기】	LEE,Gui Young
【주민등록번호】	650127-1094017
【우편번호】	139-203
【주소】	서울특별시 노원구 상계3동 107번지 불암동아아파트 105동 1005호
【국적】	KR
【발명자】	
【성명의 국문표기】	김종민
【성명의 영문표기】	KIM,Jong Min
【주민등록번호】	700103-1109233
【우편번호】	463-050
【주소】	경기도 성남시 분당구 서현동 한양아파트 333동 1005호
【국적】	KR

**【발명자】**

**【성명의 국문표기】** 인지현  
**【성명의 영문표기】** IN, Ji Hyun  
**【주민등록번호】** 771124-2063411  
**【우편번호】** 462-150  
**【주소】** 경기도 성남시 중원구 은행동 현대아파트 110동 201호  
**【국적】** KR

**【발명자】**

**【성명의 국문표기】** 정재용  
**【성명의 영문표기】** JEONG, Jae Yong  
**【주민등록번호】** 690413-1010713  
**【우편번호】** 151-054  
**【주소】** 서울특별시 관악구 봉천4동 875-1 관악캠퍼스타워 811호  
**【국적】** KR

**【발명자】**

**【성명의 국문표기】** 이동희  
**【성명의 영문표기】** LEE, Dong Hee  
**【주민등록번호】** 670210-1482310  
**【우편번호】** 151-054  
**【주소】** 서울특별시 관악구 봉천4동 875-1 관악캠퍼스타워 811호  
**【국적】** KR

**【발명자】**

**【성명의 국문표기】** 김제성  
**【성명의 영문표기】** KIM, Je Sung  
**【주민등록번호】** 680527-1024419  
**【우편번호】** 135-110  
**【주소】** 서울특별시 강남구 압구정동 464 현대아파트 61동 601호  
**【국적】** KR

**【발명자】****【성명의 국문표기】**

최종무

**【성명의 영문표기】**

CHOI, Jong Moo

**【주민등록번호】**

690120-1345439

**【우편번호】**

135-240

**【주소】**

서울특별시 강남구 개포동 주공아파트 221동 506호

**【국적】**

KR

**【심사청구】**

청구

**【취지】**

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인  
이영필 (인) 대리인  
이해영 (인)

**【수수료】****【기본출원료】**

20 면 29,000 원

**【가산출원료】**

26 면 26,000 원

**【우선권주장료】**

0 건 0 원

**【심사청구료】**

28 항 1,005,000 원

**【합계】**

1,060,000 원

**【첨부서류】**

1. 요약서·명세서(도면)\_1통

**【요약서】****【요약】**

플래시 메모리 관리방법이 개시된다.

본 발명에 따르면, 쓰기가 수행되어 소정 데이터가 기록되어 있는 페이지에 쓰기가 요청되었을 때 상기 페이지가 포함된 데이터 블록에 대응되도록 마련된 로그 블록에 쓰기를 수행하고, 상기 페이지에 쓰기를 다시 요청받으면 상기 로그 블록 내의 비어있는 자유 페이지에 쓰기를 수행한다. 이에 따르면 동일한 페이지에 연속적으로 쓰기가 요청되는 경우에도 하나의 로그 블록 내에서 처리할 수 있기 때문에 플래시 메모리 자원의 효율성을 제고할 수 있다.

**【대표도】**

도 7

**【명세서】**

**【발명의 명칭】**

플래시 메모리 관리방법{Flash memory management method}

**【도면의 간단한 설명】**

도 1은 본 발명의 바람직한 실시예에 따른 플래시 메모리 기반 시스템의 블록도,

도 2는 본 발명에 따라 플래시 메모리(1)에 마련되어 일반 데이터가 저장되기 위한 블록들을 설명하기 위한 참고도,

도 3은 로그 블록과 데이터 블록에 대한 읽기를 설명하기 위한 참고도,

도 4는 본 발명의 일 실시예에 따른 플래시 메모리(1)의 영역 분할을 설명하기 위한 참고도,

도 5는 본 발명의 다른 실시예에 따른 플래시 메모리(1)의 영역 분할을 설명하기 위한 참고도,

도 6은 로그 포인터 테이블을 설명하기 위한 참고도,

도 7은 로그 포인터 테이블 엔트리의 구조도,

도 8은 로그 포인터 테이블과 플래시 메모리(1)의 참조관계도,

도 9는 지우기-가능 블록을 설명하기 위한 참고도,

도 10은 단순 병합의 개념도,

도 11은 복사 병합의 개념도,



도 12는 본 발명에 따른 블록 병합을 수행함에 따른 블록의 전이관계를 보여주는 관계도,

도 13은 본 발명에 따른 읽기방법을 설명하기 위한 플로우차트,

도 14는 본 발명에 따른 쓰기방법을 설명하기 위한 플로우차트,

도 15는 도 14의 블록 병합을 설명하기 위한 플로우차트이다.

**【발명의 상세한 설명】**

**【발명의 목적】**

**【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<16> 본 발명은 플래시 메모리에 관한 것으로, 보다 상세하게는 플래시 메모리를 기반으로 한 시스템에 있어서 플래시 메모리 관리방법에 관한 것이다.

<17> 플래시 메모리는 전기적으로 데이터를 지우거나 쓸 수 있는 비휘발성 기억 소자이다. 마그네틱 디스크 메모리를 기반으로 한 저장장치에 비해 플래시 메모리를 기반으로 한 그것은 전력소모가 적으며 크기가 작아서 마그네틱 디스크 메모리의 대안으로서 연구 및 개발이 활발하게 진행되고 있다. 특히, 플래시 메모리는 디지털 카메라, 모바일 폰, PDA(Personal Digital Assistant)와 같은 모바일 컴퓨팅 기기를 위한 저장장치로서 각광받을 것이 예상된다.

<18> 그러나, 플래시 메모리는 데이터의 덮어쓰기가 자유로운 마그네틱 디스크 메모리와 달리 데이터의 덮어쓰기가 불가능하다. 플래시 메모리에 데이터를 덮어쓰기 위해서는 먼저 데이터를 지워야한다. 즉, 메모리 셀들을 쓰기 가능한 초기상태로 되돌려놓아야 한다. 이와 같은 작업은 지우기(erase)라고 한다. 지우

기는 일반적으로 쓰기에 비해 매우 긴 시간이 소요된다. 더욱이, 지우기는 쓰기보다 훨씬 큰 블록 단위로 수행되기 때문에 쓰기가 요청되지 않은 부분까지 함께 지워지는 결과를 초래할 수 있다. 부득이하게 지워진 부분은 다시 쓰기를 통해 복원되어야 하므로 최악의 경우 데이터 하나의 쓰기 요청이 하나의 지우기와 지워진 단위 만큼의 쓰기를 필요로 하게 된다. 이와 같이 지우기와 쓰기의 실행 단위의 불일치로 인해 쓰기의 수행 성능은 읽기의 그것에 비해 현저히 떨어지며, 심지어 기계동작으로 인한 필연적인 지연을 수반하게 되는 마그네틱 디스크 기반의 저장장치의 그것보다 낮다. 따라서 플래시 메모리를 기반으로 한 저장장치의 설계에 있어서 쓰기의 성능개선은 핵심기술에 해당된다.

<19> 미국특허 No. 5,388,083은 요청된 쓰기가 선행되어야 하는 지우기에 의해 지연되는 것을 막기 위해 지우기를 수행하지 않고 빈 공간에 쓰는 대신 사용자가 요청한 논리적 주소를 플래시 메모리 상의 물리적 주소로 변환하는 캠(CAM) 구조를 제안하고 있다. 그러나 캠 구조를 구현하기 위해서는 값비싼 회로를 필요로 한다. 미국특허 No. 5,485,595는 쓰기가 요청된 경우 지우기를 수행하지 않고 빈 공간에 쓰는 대신 각 페이지의 부가 영역에 논리적 주소를 기입하며 부가 영역에 기록된 논리적 주소를 순차적으로 검색하여 읽기를 수행하는 방식을 제안하고 있다. 그러나 이와 같은 주소변환 메커니즘은 NAND 타입의 플래시 메모리와 같이 읽기의 단위가 큰 경우에는 흩어져 저장된 주소변환 정보를 읽는데 적지 않은 시간이 소요되어 실효성이 낮은 문제점이 있다.

<20> 미국특허 No. 5,845,313은 저장장치를 초기화할 때 플래시 메모리에 저장된 논리적 주소를 스캐닝하여 별도의 RAM에 직접 주소 변환을 수행할 수 있는 선형

적인 주소변환 테이블을 구축하는 방법을 제시하고 있다. 그러나 주소변환 테이블을 저장하기 위해서는 대용량 RAM이 필요하다. 예를 들어, 전체 저장용량이 32MB이고 페이지의 크기가 512B인 플래시 메모리 기반 저장장치의 주소변환 테이블을 저장하기 위해서는 65,536개의 페이지 당 2B가 필요하므로 총 128KB의 RAM이 필요하게 된다. 이러한 요구량은 모바일 기기 등 자원이 귀한 소규모 시스템에 있어서는 지나치게 크다.

<21> 미국특허 No. 5,404,485는 쓰기를 위해 새로운 블록(교체 블록)을 할당하고 할당된 블록에 쓰기를 수행하는 방식을 제안하고 있다. 그러나, 이에 따르면 쓰기를 위해 계속 새로운 블록이 할당되므로 동일한 페이지가 기록된 서로 다른 버전의 복수개의 블록이 존재하게 된다. 즉, 묵시적으로 모든 블록마다 적어도 하나의 교체 블록이 존재할 것을 요구하여 플래시 메모리의 용량이 상당히 줄어들게 된다. 또한, 새로운 블록에 쓰여지는 페이지는 항상 이전 블록의 그것과 동일한 위치에 쓰여져야 하므로, 특정 페이지에 대해서만 갱신이 이루어지고 나머지 페이지는 갱신이 거의 없는 경우 특정 페이지의 내용만 상이할 뿐 나머지 페이지의 내용은 동일한 교체 블록이 복수개 존재하게 되어 플래시 메모리의 저장공간의 낭비가 심하게 되는 문제점이 있다. 이에, 모바일 기기 등 소규모 시스템에 적합하지 않다.

**【발명이 이루고자 하는 기술적 과제】**

<22> 따라서, 본 발명의 목적은 플래시 메모리의 성능을 개선시킬 수 있는 플래시 메모리 기반 시스템 및 그 관리방법을 제공하는 것이다.

<23> 본 발명의 다른 목적은 전원 차단 등의 비상시에 일관성있는 데이터의 복원이 가능한 플래시 메모리 기반 시스템 및 그 관리방법을 제공하는 것이다.

<24> 본 발명의 또 다른 목적은 FAT(File Allocation Table) 기반으로 한 DOS 파일시스템에서와 같이 특정 페이지에 대한 데이터 갱신이 잦은 환경에서도 시스템의 성능이 저하되지 않는 플래시 메모리 기반 시스템 및 그 관리방법을 제공하는 것이다.

**【발명의 구성 및 작용】**

<25> 상기 목적은 플래시 메모리에 소정 데이터를 쓰는 방법에 있어서, (a) 쓰기가 수행되어 소정 데이터가 기록되어 있는 페이지에 쓰기를 요청받는 단계; (b) 상기 페이지가 포함된 데이터 블록에 대응되도록 마련된 로그 블록에 쓰기를 수행하는 단계; (c) 상기 페이지에 쓰기를 다시 요청받는 단계; 및 (d) 상기 로그 블록 내의 비어있는 자유 페이지에 쓰기를 수행하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법에 의해 달성된다.

<26> 상기 (b)단계는 (b11) 비어있는 자유 페이지에 쓰기를 수행하는 단계를 포함하거나, (b21) 상기 로그 블록을 할당하는 단계; 및 (b22) 상기 쓰기가 요청된 페이지의 상기 데이터 블록에서의 위치와 동일한 위치의 자유 페이지에 쓰기를 수행하는 단계를 포함하는 것이 바람직하다.

<27> 또한, 플래시 메모리에 소정 데이터를 쓰는 방법에 있어서, (a) 소정 페이지에 쓰기를 요청받는 단계; (b) 상기 페이지가 포함된 제1 데이터 블록에 대응되는 제1-1 로그 블록을 할당하는 단계; (c) 상기 제1-1 로그 블록 내의 비어있

는 자유 페이지에 쓰기를 수행하는 단계; (d) 상기 페이지에 쓰기를 다시 요청 받는 단계; 및 (e) 상기 제1-1 로그 블록 내의 비어있는 자유 페이지에 쓰기를 수행하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법에 의해서도 달성된다.

<28>      상기 (b)단계는 (b1) 제2 데이터 블록 및 이에 대응되는 제2 로그 블록을 기초로 제3 데이터 블록을 생성하는 블록 병합을 수행하는 단계; 및 (b2) 상기 제2 데이터 블록에 대한 지우기를 수행하여 얻어진 자유 블록을 상기 제1 로그 블록으로 할당하는 단계를 포함하는 것이 바람직하다. 상기 (b1)단계는 상기 제1-1 로그 블록을 할당하기 위한 자유 블록이 존재하지 않을 때 수행되거나, 상기 제1 데이터 블록에 대응되는 기존의 로그 블록이 모두 사용 중일 경우 수행되는 것이 바람직하다.

<29>      또한, 상기 (b1)단계는 (b11) 상기 제2 로그 블록의 페이지와 상기 제2 데이터 블록의 페이지의 배열순서가 동일하고 일대일 대응되는 경우 상기 제2 로그 블록을 상기 제3 데이터 블록으로 전이시키는 교환 병합을 수행하는 단계를 포함하거나,

<30>      (b12) 상기 제2 로그 블록에 존재하는 페이지들이 모두 한번씩만 쓰기 요청된 경우 상기 제2 로그 블록의 자유 페이지에 상기 제2 데이터 블록의 해당 페이지를 복사하여 상기 제3 데이터 블록을 생성하는 복사 병합을 수행하는 단계를 포함하거나,

<31>      (b13) 데이터가 기록되어 있지 않은 자유 블록에 상기 제2 로그 블록에 존재

하는 최신 페이지들을 복사하고, 나머지 자유 페이지에 상기 제2 데이터 블록의 해당 페이지를 복사하여 상기 제3 데이터 블록을 생성하는 단순 병합을 수행하는 단계를 포함하는 것이 더욱 바람직하다.

<32>       상기 (e)단계는 (e1) 상기 제1-1 로그 블록 내에 자유 페이지가 존재하지 않을 경우 새로운 제1-2 로그 블록을 할당하는 단계; 및 (e2) 상기 제1-2 로그 블록 내의 자유 페이지에 쓰기를 수행하는 단계를 포함하고, 상기 (e1)단계는 (e11) 상기 제1-1 로그 블록의 페이지와 상기 제1 데이터 블록의 페이지의 배열 순서가 동일하고 일대일 대응되는 경우 상기 제1-1 로그 블록을 제2 데이터 블록으로 전이시키는 교환 병합을 수행하는 단계; 및 (e12) 상기 제1 데이터 블록에 대한 지우기를 수행하여 얻어진 자유 블록을 상기 제1-2 로그 블록으로 할당하는 단계를 포함하거나,

<33>       (e21) 상기 제1-1 로그 블록에 존재하는 페이지들이 모두 한번씩만 쓰기 요청된 경우 상기 제1-1 로그 블록의 자유 페이지에 상기 제1 데이터 블록의 해당 페이지를 복사하여 제2 데이터 블록을 생성하는 복사 병합을 수행하는 단계; 및 (e22) 상기 제1 데이터 블록에 대한 지우기를 수행하여 얻어진 자유 블록을 상기 제1-2 로그 블록으로 할당하는 단계를 포함하거나,

<34>       (e31) 자유 블록에 상기 제1-1 로그 블록에 존재하는 최신 페이지들을 복사하고, 나머지 자유 페이지에 상기 제1 데이터 블록의 해당 페이지를 복사하여 제2 데이터 블록을 생성하는 단순 병합을 수행하는 단계; 및 (e32) 상기 제1 데이터 블록 또는 상기 제1-1 로그 블록에 대한 지우기를 수행하여 얻어진 자유 블

록을 상기 제1-2 로그 블록으로 할당하는 단계를 포함하는 것이 특히 바람직하다

<35>      상기 (e2)단계는 (e21) 쓰기가 요청된 페이지의 상기 데이터 블록에서의 위치와 동일한 위치의 자유 페이지에 쓰기를 수행하는 단계를 포함하는 것이 바람직하다.

<36>      한편, 본 발명의 다른 분야에 따르면, 상기 목적은 플래시 메모리로부터 소정 데이터를 읽는 방법에 있어서, (a) 로그 포인터 테이블에 요청된 페이지의 논리적 주소 중 블록 주소 부분이 기록된 엔트리를 검색하는 단계; (b) 검색된 엔트리에 상기 요청된 페이지의 논리적 주소가 기록되어 있는지 여부를 확인하는 단계; 및 (c) 검색된 엔트리에 기록된 대응 로그 블록의 물리적 주소와, 검색된 논리적 주소의 상기 검색된 엔트리에서의 기록 위치를 참조하여 상기 로그 블록의 해당 페이지에 접근하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 읽기방법에 의해서도 달성된다.

<37>      상기 (c)단계는 상기 로그 블록의 해당 페이지에 접근 함에 있어 상기 검색된 논리적 주소의 상기 검색된 엔트리에서의 기록 위치와 동일한 위치의 페이지에 접근하는 것이 바람직하다.

<38>      한편, 본 발명의 다른 분야에 따르면, 상기 목적은, 데이터 블록, 및 상기 데이터 블록을 갱신하기 위한 데이터를 쓰기 위한 로그 블록을 포함하는 플래시 메모리를 관리하는 방법에 있어서, (a) 제1 데이터 블록의 페이지와 상기 제1 데이터 블록에 대응되는 제1 로그 블록의 페이지의 배열순서가 동일하고 일대일 대응되는 경우 상기 제1 로그 블록을 제2 데이터 블록으로 전이시키는 단계; 및

(b) 주소변환 정보를 갱신하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 관리방법에 의해서도 달성된다.

<39> 또한, 데이터 블록, 및 상기 데이터 블록을 갱신하기 위한 데이터를 쓰기 위한 로그 블록을 포함하는 플래시 메모리를 관리하는 방법에 있어서, (a) 제1 로그 블록에 존재하는 페이지들이 모두 한번씩만 쓰기 요청된 경우 상기 제1 로그 블록의 자유 페이지에 대응 제1 데이터 블록의 해당 페이지를 복사하여 제2 데이터 블록을 생성하는 단계; 및 (b) 주소변환 정보를 갱신하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 관리방법에 의해서도 달성된다.

<40> 또한, 데이터 블록, 및 상기 데이터 블록을 갱신하기 위한 데이터를 쓰기 위한 로그 블록을 포함하는 플래시 메모리를 관리하는 방법에 있어서, (a) 데이터가 기록되어 있지 않은 자유 블록에 제1 로그 블록에 존재하는 최신 페이지들을 복사하고, 나머지 자유 페이지에 대응 제1 데이터 블록의 해당 페이지를 복사하여 제2 데이터 블록을 생성하는 단계; 및 (b) 주소변환 정보를 갱신하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 관리방법에 의해서도 달성된다.

<41> 상기 (a)단계 이전에 (a0) 상기 (a)단계 또는 (b)단계의 수행 중 시스템이 중단되는 경우 데이터를 복구하기 위한 복구 정보를 기록하는 단계를 더 포함하는 것이 바람직하다.

<42> 상기 플래시 메모리 관리방법은 (c) 상기 (a)단계 또는 (b)단계의 수행 중 시스템이 중단되는 경우 상기 복구 정보를 참조하여 데이터를 복원하는 단계를 더 포함하는 것이 바람직하다.



<43>      상기 복구 정보는 상기 자유 블록의 리스트, 로그 블록의 리스트, 상기 로그 블록을 관리하기 위한 데이터 구조인 로그 포인터 테이블을 포함하며, 상기 로그 포인터 테이블에는 로그 블록에 대응되는 갯수의 로그 포인터 테이블 엔트리가 구성되어 있고, 각 엔트리에는 대응 데이터 블록의 논리적 주소와 해당 로그 블록의 물리적 주소가 매핑되어 있으며, 해당 로그 블록 내의 각 페이지들의 물리적 배열 순서에 따라 해당 데이터 블록의 요청된 페이지의 논리적 주소가 기록되어 있다.

<44>      또한, 상기 목적은, 데이터 블록, 및 상기 데이터 블록을 갱신하기 위한 데이터를 쓰기 위한 로그 블록을 포함하는 플래시 메모리를 관리하는 방법에 있어서, (a) 플래시 메모리의 소정 영역을 할당하고 할당된 영역에 복구 정보로서 상기 데이터 블록 및 상기 로그 블록의 리스트, 및 상기 로그 블록을 관리하기 위한 데이터 구조를 기록하는 단계; (b) 시스템이 중단되는 경우 상기 복구 정보를 기초로 현재 플래시 메모리에 기록된 상태를 점검하여 오류 발생 여부를 확인하는 단계; 및 (c) 오류가 발생된 경우 상기 복구 정보를 참조하여 데이터를 복원하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리 관리방법에 의해서도 달성된다.

<45>      이하, 첨부된 도면을 참조하여 본 발명 바람직한 실시예를 설명한다.

<46>      도 1은 본 발명의 바람직한 실시예에 따른 플래시 메모리 기반 시스템의 블록도이다.

- <47> 도 1을 참조하면, 시스템은 플래시 메모리(1), ROM(2), RAM(3) 및 프로세서(4)를 구비한다. 프로세서(4)는 통상 ROM(2)에 기록되어 있는 프로그램 코드와 결합되어 플래시 메모리(1) 또는 RAM(2)에 대한 일련의 읽기 또는 쓰기 명령을 발한다. 플래시 메모리(1)에는 본 발명의 플래시 메모리 관리방법에 따른 쓰기와 읽기가 수행된다. ROM(2)과 RAM(3)에는 프로세서(4)에서 수행되는 응용 프로그램 코드 또는 관련 데이터 구조가 저장된다.
- <48> 도 2는 본 발명에 따라 플래시 메모리(1)에 마련되어 일반 데이터가 저장되기 위한 블록들을 설명하기 위한 참고도이다.
- <49> 도2를 참조하면, 플래시 메모리(1)에는 복수개의 데이터 블록과 적어도 일부의 데이터 블록에 대응되도록 마련되는 로그 블록이 존재한다. 「데이터 블록」은 일반 데이터를 저장하는 블록을 가리키며, 「로그 블록」은 데이터 블록의 소정 부분을 수정하고자 할 경우에 할당되어 수정 내용을 기록하는 용도로 활용되는 블록을 가리킨다. 따라서, 로그 블록은 오직 하나의 데이터 블록에 대응되며 해당 데이터 블록의 수정된 페이지들을 저장한다. 로그 블록에 저장되어 있는 페이지들은 데이터 블록에 저장되어 있는 해당 페이지보다 우선적으로 참조된다. 이처럼 우선적으로 참조되는 페이지를 본 명세서에서는 「유효 페이지」라고 한다. 더불어, 물리적으로 유효한 데이터가 기록되어 있더라도 유효 페이지에 의해 그 내용이 무시되는 페이지는 논리적인 의미에서 「무효 페이지」라고 부른다.
- <50> 도 3은 로그 블록과 데이터 블록에 대한 읽기를 설명하기 위한 참고도이다.

<51> 도 3을 참조하면, 사용자로부터 논리적 주소(Logical address)와 함께 소정 페이지에 대한 읽기가 요청되면 프로세서(4)는 RAM(3)에 기록되어 있는 로그 포인터 테이블을 참조하여 해당 로그 블록이 존재하는지 여부를 확인하고, 존재할 경우 그 로그 블록에 요청된 페이지가 유효하게 저장되어 있는지 여부를 확인한다. 요청된 페이지가 유효하게 저장되어 있으면 로그 블록의 해당 페이지를 읽고 그렇지 않으면 해당 데이터 블록에 기록된 해당 페이지를 읽는다. 로그 포인터 테이블은 후술한다.

<52> 도 4는 본 발명의 일 실시예에 따른 플래시 메모리(1)의 영역 분할을 설명하기 위한 참고도이다. 도 4를 참조하면, 플래시 메모리(1)는 맵 영역, 로그 블록 영역, 데이터 블록 영역, 및 자유 블록 영역으로 분할될 수 있다. 맵 영역에는 주소변환 정보가 저장되고, 로그 블록 영역은 로그 블록으로 할당되기 위한 영역이며, 데이터 블록 영역은 일반 데이터가 기록되는 영역이고, 자유 블록 영역은 로그 블록 또는 데이터 블록으로 할당하기 위한 영역이다. 여기서, 각 영역은 논리적으로 분할된 영역을 의미한다. 따라서, 물리적으로는 서로 뒤섞여서 불연속적으로 존재할 수 있다. 특히, 데이터 블록 영역, 로그 블록 영역, 및 자유 블록 영역은 그러하다.

<53> 도 5는 본 발명의 다른 실시예에 따른 플래시 메모리(1)의 영역 분할을 설명하기 위한 참고도이다. 도 5를 참조하면, 플래시 메모리(1)는 맵 영역, 체크 포인트 영역, 로그 블록 영역, 데이터 블록 영역, 및 자유 블록 영역으로 분할될 수 있다. 본 실시예에서는 도 4의 영역 분할에 비해 체크 포인트 영역이 새롭게 마련되어 있다. 체크 포인트 영역에는 데이터 복구를 위해 필요한 복구 정보가

기록된다. 한편, 도 4의 경우와 마찬가지로, 맵 영역에는 주소변환 정보가 저장되고, 로그 블록 영역은 로그 블록으로 할당되기 위한 영역이며, 데이터 블록 영역은 일반 데이터가 기록되는 영역이고, 자유 블록 영역은 로그 블록 또는 데이터 블록으로 할당하기 위한 자유 영역이다. 맵 영역에 기록되는 주소변환 정보 및 체크 포인트 영역에 기록되는 복구 정보에 대한 상세한 설명은 후술한다.

<54> 「로그 포인터 테이블」은 로그 블록의 관리를 위한 데이터 구조이다. 로그 포인터 테이블에는 대응 데이터 블록의 논리적 주소, 해당 로그 블록의 물리적 주소, 및 로그 블록 내의 각 페이지들의 물리적 배열 순서에 따라 해당 데이터 블록의 갱신된 페이지의 오프셋값(요청된 페이지의 논리적 주소)이 기록된다. 로그 포인터 테이블은 본 실시예에 따라 RAM(3)에 구축되어야 하므로 프로세서(4)가 로그 블록 영역을 스캔하여 구성한다. 도 6을 참조하면, 로그 포인터 테이블에는 로그 블록에 대응되는 갯수의 로그 포인터 테이블 엔트리가 배치되어 있다. 해당 페이지의 논리적 주소와 함께 쓰기 또는 읽기 명령이 요청되면 프로세서(4)는 로그 포인터 테이블을 참조하여 해당 엔트리의 존재여부에 따라 로그 블록 또는 데이터 블록에 접근하게 된다.

<55> 도 7은 로그 포인터 테이블 엔트리의 구조도이다.

<56> 도 7을 참조하면, 로그 포인터 테이블 엔트리에는 데이터 블록의 논리적 주소(log\_blk)와 해당 로그 블록의 물리적 주소(phy\_blk)가 기록되어 있다. 또한, 데이터 블록의 페이지가 기록된 순서대로 해당 로그 블록 내의 대응 페이지의 논리적 주소 page #0, page #1, ..., page #N가 기록되어 있다.

<57> 예를 들면, 1개의 블록에 16개의 페이지가 존재하는 경우 논리적 주소가 02FF(16진수)라면, 앞의 세자리 02F는 블록 주소를 의미하고, 마지막 한자리 F는 해당 블록에 존재하는 페이지의 오프셋값이 된다. 따라서, 로그 포인터 테이블의 log\_blk 중 02F가 있는지 여부를 검색하여 대응 로그 블록의 존재여부를 확인할 수 있다. 대응 로그 블록이 존재하면 요청된 페이지의 논리적 주소 02FF, 또는 오프셋값 F가 기록되어 있는지 여부를 확인하여 해당 로그 블록 내의 갱신된 페이지의 위치를 알 수 있다. 예를 들어, 만약 page #0가 F라면 로그 블록 내의 첫 번째 물리적 페이지에 요청된 페이지가 기록되어 있다.

<58> 이처럼, 요청된 논리적 주소 중 일부분, 즉 블록 주소 부분만을 사용하여 로그 블록의 존재 여부를 확인하여 해당 로그 블록에 접근하는 것을 「블록 어드레싱」이라 정의하고, 요청된 논리적 주소 전체(또는 오프셋값)를 사용하여 해당 로그 블록의 페이지에 액세스하는 것을 「페이지 어드레싱」으로 정의한다. 이처럼, 본 발명은 블록 어드레싱과 페이지 어드레싱을 함께 채용하여 동일한 페이지가 여러번 갱신되더라도 하나의 로그 블록에 이를 모두 기록할 수 있다.

<59> 도 8은 로그 포인터 테이블과 플래시 메모리(1)의 참조관계도이다.

<60> 도 8에 도시된 바와 같이, log\_blk를 참조하여 해당 데이터 블록에 대한 로그 블록이 존재하는지를 검색할 수 있으며, phy\_blk를 참조하여 대응 로그 블록이 기록된 위치를 알 수 있다. 또한, 로그 포인터 테이블 엔트리에는 본 실시예에 따라 해당 로그 블록의 각 페이지의 논리적 주소 page #0, page #1, ..., page #15가 기록될 수 있다. 본 실시예에서의 각 블록에는 16개의 페이지가 포함된다.

- <61> 원칙적으로 갱신된 페이지들은 로그 블록에 기록됨에 있어 해당 데이터 블록의 대응 페이지와 동일한 위치에 기록된다. 실제로, 로그 블록에 갱신된 페이지가 처음 기록되는 경우에는 데이터 블록의 해당 페이지와 동일한 위치에 기록될 수 있다. 그러나, 이후에는 반드시 그런 것은 아니다. 즉, 해당 데이터 블록의 소정 페이지가 나머지 페이지가 한번씩 갱신되기 이전에 재차 갱신되어야 하는 경우에는 해당 로그 블록의 나머지 빈 공간에 기록된다.
- <62> 도 9는 지우기-가능 블록을 설명하기 위한 참고도이다.
- <63> 데이터 블록의 모든 페이지들이 모두 한번씩만 갱신되면 로그 블록의 페이지들은 도 8에 도시된 바와 같이, 해당 데이터 블록의 각 페이지들에 일대일 대응된다. 이같은 경우에는 로그 블록이 해당 데이터 블록의 모든 내용을 포함하고 있으므로 해당 데이터 블록을 지우더라도 데이터의 손실은 발생되지 않는다. 이처럼 더 이상 유효한 데이터를 가지고 있지 않은(완전히 새도우된) 데이터 블록은 「지우기-가능(erasable) 블록」이라고 부른다. 지워진 블록은 「자유(free) 블록」이라 한다. 지우기-가능 블록은 언제라도 지울 수 있으며 자유 블록은 필요에 따라 데이터 블록 또는 로그 블록으로 할당할 수 있다.
- <64> 한편, 본 발명에서는 「블록 병합(block merge)」이 수행된다. 블록 병합은 원칙적으로 쓰기가 반복되어 로그 블록에 쓰기 가능한 페이지가 존재하지 않게 되면 이루어진다. 이같은 경우, 블록 병합을 통해 로그 블록과 해당 데이터 블록을 합하여 새로운 데이터 블록을 생성하고 기존의 로그 블록은 지워서 자유 블록으로 만들어준다.

- <65>      특히, 데이터 블록의 모든 페이지들이 모두 한번씩만 갱신되어 로그 블록과 데이터 블록의 페이지 배열이 동일한 경우 수행되는 블록 병합은 「교환 병합 (switch merge)」이라고 부른다.
- <66>      로그 블록의 페이지와 해당 데이터 블록의 페이지의 배열이 일치하지 않을 경우에는 「단순 병합(simple merge)」이 이루어진다. 나아가, 로그 블록이 현재 모두 사용 중이기 때문에 새로 요청된 쓰기를 수행하기 위해 로그 블록이 다시 할당되어야 할 경우에도 수행된다. 이 같은 경우에는 병합 대상이 되는 로그 블록이 아직 사용되지 않은 자유 페이지를 가지고 있을 수 있다.
- <67>      만약 로그 블록에 존재하는 페이지들이 단 한번씩만 갱신된 경우라면 비어 있는 페이지들을 데이터 블록의 해당 페이지들로 채움으로써 로그 블록을 데이터 블록으로 전이시킬 수 있으며, 이러한 블록 병합을 「복사 병합(copy merge)」이라고 한다. 정리하면, 블록 병합은 교환 병합, 단순 병합, 및 복사 병합의 세 가지가 있다.
- <68>      교환 병합은 도 9를 참조하여 전술한 바와 같이 데이터 블록의 모든 페이지들이 모두 한번씩만 갱신된 로그 블록을 그대로 데이터 블록으로 전이시킴으로써 수행된다. 블록의 전이는 데이터 블록 또는 로그 블록에 기록된 데이터의 복사 없이 주소변환 정보만을 갱신함으로써 이루어진다. 즉, 맵 영역에 기록된 주소변환 정보를 갱신하여 사용자가 요청한 논리적 주소에 대해 해당 로그 블록이 매핑되도록 한다. 맵 영역에는 블록 어드레싱을 가능하게 하기 위해 각 블록마다의 주소변환 정보가 기록되어 있다. 여기서, 무효 페이지는 전술한 바와 같이

유효 페이지로 인해 그 내용이 무시되는 페이지라는 의미로 사용된 것이며 실제 구현에 있어서는 물리적으로 유효한 페이지일 수 있다.

<69> 단순 병합에 의해, 도 10에 도시된 바와 같이, 새로운 자유 블록의 동일한 위치에 데이터 블록의 유효 페이지 및 해당 로그 블록의 유효 페이지를 기록하여 새로운 데이터 블록이 생성된다. 이에 기존의 데이터 블록과 로그 블록은 지우기-가능 블록이 된다.

<70> 복사 병합은 도 11에 도시된 바와 같이, 기존의 데이터 블록에 기록된 유효 페이지를 로그 블록의 자유 페이지에 복사함으로써 이루어진다. 기존의 데이터 블록은 지우기-가능 블록으로 전이된다. 블록 병합에 있어서 지칭되는 무효 페이지는 전술한 바와 같이 맨 먼저 참조되지 않는 페이지라는 의미로 사용된 것이며 실제 구현에 있어서는 물리적으로 유효한 페이지일 수 있다.

<71> 도 12는 본 발명에 따른 블록 병합을 수행함에 따른 블록의 전이관계를 보여주는 관계도이다.

<72> 도 12를 참조하면, 자유 블록은 로그 블록 또는 데이터 블록으로 전이된다. 로그 블록은 교환 병합 또는 복사 병합을 통해 데이터 블록으로 전이되거나 단순 병합을 통해 지우기-가능 블록으로 전이된다. 데이터 블록은 교환 병합, 단순 병합 및 복사 병합을 통해 지우기-가능 블록으로 전이될 수 있다. 지우기-가능 블록은 지워짐으로써 다시 자유 블록으로 전이된다.

<73> 블록 병합을 수행하기 위해서는 플래시 메모리(1)에 존재하는 자유 블록과 지우기-가능 블록에 대한 리스트를 가지고 있어야 한다. 자유 블록과 지우기-가



능 블록의 리스트는 RAM(3)에 로그 포인터 테이블과 함께 기록되는 데이터 구조로서, 맵 영역 또는 체크 포인트 영역에 기록될 수 있다.

<74> 자유 블록의 리스트, 지우기-가능 블록의 리스트, 및 로그 포인터 테이블은 시스템이 초기화될 때 RAM(3)에 복원되어야 한다. 체크 포인트 영역은 이들의 신속하고 완전한 복구를 위해 필요한 복구 정보가 기록되기 위해 본 발명의 일 실시예에 따라 할당된 영역이다. 체크 포인트 영역이 설정되는 경우, 여기에는 복구 정보로서 앞서 언급한 바 있는 자유 블록의 리스트, 지우기-가능 블록의 리스트, 및 로그 블록의 리스트가 저장된다. 특히, 체크 포인트 영역에는 블록 병합의 수행 중 시스템의 폭주나 갑작스러운 전원 차단 등으로 정보가 손상되는 것을 방지하기 위해, 블록 병합을 수행하기 이전에 어떤 블록 병합을 수행할 것이며 그 결과 어떤 블록이 어떤 블록으로 전이될 것인지를 기술하는 플랜 로그(plan log)가 저장된다. 플랜 로그에는 수행하고자 하는 블록 병합의 종류, 자유 블록에서 데이터 블록으로 전이되는 블록, 자유 블록에서 로그 블록으로 전이되는 블록, 데이터 블록에서 자유 블록으로 전이되는 블록, 로그 블록에서 자유 블록으로 전이되는 블록의 물리적 주소가 기록된다. 체크 포인트 영역에는 주소 변환 정보의 구축을 위해 필요한 정보(예를 들면 주소변환 정보가 저장된 위치)도 기록된다. 체크 포인트 영역 자체의 위치는 플래시 메모리(1)의 미리 정해진 블록에 기록된다.

<75> 상기와 같은 구성에 의해 본 발명의 바람직한 실시예에 따른 플래시 메모리 관리방법을 설명하면 다음과 같다. 이하에서는 본 발명에 따른 플래시 메모리

관리방법을 시스템 초기화에 수반되는 데이터 구조 구축 및 복구 방법, 읽기방법 및 쓰기방법으로 나누어 기술한다.

<76> 먼저 시스템 초기화에 수반되는 플래시 메모리 관리방법은 데이터 구조 구축 및 복구 방법을 의미한다. 즉, 쓰기 및 읽기에 필요한 데이터 구조(자유 블록의 리스트, 지우기-가능 블록의 리스트, 로그 블록의 리스트 및 로그 포인터 테이블) 및 주소변환 정보를 구축하고 구축된 정보의 일관성(integrity) 검사를 수행하여 복구 작업이 필요한 경우 복구 정보를 바탕으로 복구 작업을 수행하는 것을 의미한다. 도 1의 시스템이 초기화되면 프로세서(4)는 RAM(3)에 로그 포인터 테이블, 자유 블록의 리스트, 지우기-가능 블록의 리스트, 및 로그 블록의 리스트를 구축하여야 한다. 이를 위해 프로세서(4)는 플래시 메모리(1)의 체크 포인트 영역의 가장 최근에 기록된 페이지로부터 복구 정보를 읽어들인다. 복구 정보가 순차적으로 기록되는 경우에는 체크 포인트 영역에서 첫 번째로 발견되는 자유 페이지(비어있는 페이지)의 직전에 위치한 페이지에 최신의 복구 정보가 기록되기 때문이다. 다만, 전술한 가장 최근에 기록된 페이지의 식별이 가능한 한 복구 정보의 기록순서는 필요에 따라 변경할 수 있다.

<77> 로그 포인터 테이블은 복구 정보에 지정된 로그 블록들의 모든 페이지들을 스캐닝하여 매 페이지마다 부가되어 있는 영역에 저장된 논리적 주소를 읽어냄으로써 구축할 수 있다. 맵 영역에서도 주소변환 정보는 순차적으로 기록되므로 마지막으로 기록된 페이지(첫 번째 자유 페이지의 직전에 위치한 페이지)를 최종 변경된 것으로 간주하여 이를 기초로 주소변환 정보를 구축할 수 있다. 자유 블

록 리스트와 지우기-가능 블록 리스트는 또한 복구 정보를 토대로 그대로 복구가 가능하다.

<78> 다음으로 플랜 로그를 참조하여 구축한 정보(자유 블록의 리스트, 지우기-가능 블록의 리스트, 로그 블록의 리스트 및 로그 포인터 테이블)를 검증한다. 다시 말해, 블록 병합을 수행하는 도중 시스템 동작이 중단된 경우 등이 발생되어 구축한 정보가 실제 상황과 맞는지 여부를 검사하여야 한다. 구체적으로, 시스템 동작이 중단되는 경우는 첫째, 체크 포인트 영역에 복구 정보를 쓰는 중, 둘째 블록 병합을 수행하는 중, 셋째 맵 영역에 주소변환 정보를 갱신하는 중, 넷째 지우기 중의 네 가지로 분류할 수 있으며, 각각의 경우 구축된 정보가 실제 상황과 맞는지 여부를 검사하여 맞지 않을 경우 복구하는 방법은 다음과 같다.

<79> 1. 체크 포인트 영역에 복구 정보를 쓰는 중에 시스템 동작이 중단된 경우: 체크 포인트 영역의 첫 번째 자유 페이지를 찾아 이 페이지가 실제로 자유 페이지인지 아닌지 여부를 데이터를 읽어 확인한다. 만약 자유 페이지가 비어있지 않다면 체크 포인트 영역에 복구 정보를 쓰는 중에 시스템의 동작이 중단된 것으로 판단할 수 있다. 이같은 경우는 실제 데이터가 쓰기가 수행되기 전이므로 별개의 복구절차를 밟을 필요가 없으며 다만 최종적으로 기록된 복구 정보를 무시한다.

<80> 2. 블록 병합을 수행하는 중에 시스템 동작이 중단된 경우: 플랜 로그에 데이터 블록으로 전이될 것으로 기재된 블록의 모든 페이지에 제대로 데이터가 기록되어 있는지(유효한지) 여부를 검사한다. 만일 하나의 페이지라도 유효하지

않다면 블록 병합 도중 시스템 동작이 중단된 것으로 판단할 수 있다. 이와 같은 경우에는 블록 병합을 다시 수행함으로써 데이터를 올바르게 복구할 수 있다.

<81> 3. 주소변환 정보의 갱신 중에 시스템 동작이 중단된 경우: 플랜 로그에 데이터 블록으로 전이될 것으로 기록된 블록으로부터 논리적 주소를 읽어들이며 맵 영역에 기록된 정보와 일치하는지 여부를 확인한다. 일치하지 않으면 주소변환 정보를 갱신하는 중에 시스템 동작이 중단된 것으로 볼 수 있다. 이같은 경우에는 데이터 블록으로부터 얻은 논리적 주소와 해당 물리적 주소를 바탕으로 주소 변환 정보를 수정함으로써 복원이 가능하다.

<82> 4. 지우기 중에 시스템 동작이 중단된 경우: 플랜 로그에 자유 블록으로 전이될 예정으로 기록된 블록들이 실제 자유 블록인지 여부를 검사한다. 자유 블록이 아니라면(페이지가 전부 비어 있지 않다면) 비어있지 않은 블록에 대해 지우기를 다시 수행한다.

<83> 시스템 초기화에 수반되는 플래시 메모리 관리방법을 통해 전술한 바와 같이 필요한 데이터 구조를 구축하고 일관성 검사까지 완료되면 읽기 및 쓰기를 할 수 있다.

<84> 도 13은 본 발명에 따른 읽기방법을 설명하기 위한 플로우차트이다.

<85> 이해를 돕기 위해 읽기방법을 개관하면, 프로세서(4)는 요청된 페이지가 로그 블록에 존재하는지 여부를 검색하고, 검색된 로그 블록으로부터 해당 페이지를 읽어온다.

- <86> 보다 구체적으로, 먼저 프로세서(4)는 요청된 페이지의 논리적 주소를 기초로 로그 포인터 테이블을 순차적으로 검색하여 부합하는 엔트리가 있는지 살펴본다(1301단계). 요청된 페이지의 논리적 주소는 블록 어드레싱 부분과 페이지 어드레싱 부분으로 결합되어 있기 때문에 블록 어드레싱 부분을 참조하여 엔트리를 검색한다. 부합되는 엔트리가 발견되면(1302단계), 발견된 엔트리를 검색하여 요청된 페이지가 존재하는지 살펴본다(1303단계)
- <87> 해당 페이지가 발견되면 이를 읽는다(1305단계). 이 때, 동일한 페이지가 둘 이상 발견되면 동일한 오프셋값을 갖는 것을 제외하고 가장 마지막으로 발견된 것을 최신의 것으로 간주하여 로그 블록의 해당 페이지를 읽는다. 부합되는 엔트리가 발견되지 않거나(1302단계), 대응 로그 블록에 해당 페이지가 존재하지 않으면(1304단계) 요청된 논리적 주소를 기초로 데이터 블록 내의 해당 페이지를 읽는다(1306단계).
- <88> 도 14는 본 발명에 따른 쓰기방법을 설명하기 위한 플로우차트이다.
- <89> 이해를 돕기 위해 쓰기방법을 개관하면, 프로세서(4)는 요청된 페이지가 로그 블록에 존재하는지 여부를 먼저 검색하고, 로그 블록이 존재할 경우 로그 블록에 요청된 페이지와 동일한 위치의 페이지가 사용가능한 상태인지 여부를 검사한다. 사용가능하면 해당 페이지에 쓰기를 수행하며, 그렇지 않은 경우에는 로그 블록의 사용가능한 다른 페이지에 쓰기를 수행하며, 로그 블록에 사용가능한 페이지가 존재하지 않으면 로그 블록을 새롭게 할당하여 동일한 위치에 쓰기를 수행한다.

<90> 보다 구체적으로, 프로세서(4)는 요청된 페이지의 논리적 주소에 기초하여 로그 포인터 테이블을 검색한다(1401단계). 엔트리가 발견되면(1402단계), 해당 로그 블록이 존재한다는 것을 의미하므로 엔트리를 검색하여 요청된 페이지와 동일한 오프셋값을 갖는 페이지가 사용가능한지 여부를 판단하고(1403단계), 사용 가능하면 해당 페이지에 쓰기를 수행한다(1404단계). 여기서, 사용가능한 페이지라 함은 로그 블록의 해당 페이지에 아직 쓰기가 수행되지 않아 비어있는 페이지(자유 페이지)를 말한다. 자유 페이지가 존재하는지 여부는 엔트리를 검색하여 해당 페이지가 유효한지 여부(우선적으로 참조되는지, 즉 데이터가 기록되어 있는지)로 판단할 수 있다. 다음으로, 로그 포인터 테이블의 해당 엔트리에 쓰기가 수행된 페이지의 논리적 주소에 대응하는 물리적 주소를 기입한다. 이 경우 사용자의 쓰기 요청은 플래시 메모리에 대한 하나의 쓰기만으로 완료된다.

<91> 대응 로그 블록은 발견되었으나 오프셋값이 같은 페이지가 이미 사용되었다면(1403단계), 로그 블록 내에 다른 자유 페이지가 할당될 수 있는지 확인하고(1406단계), 할당된 자유 페이지에 쓰기를 수행한다(1407단계). 두 개 이상의 자유 페이지가 존재한다면 로그 블록의 시작 지점부터 순차적으로 검색하여 가장 가까운 페이지를 할당할 수 있다. 이어서, 로그 포인터 테이블의 대응 엔트리에 요청된 페이지의 논리적 주소에 대응되도록, 할당된 페이지의 물리적 주소를 기록한다(1405단계).

<92> 로그 포인터 테이블을 검색한 결과 요청된 페이지에 대한 대응 엔트리가 발견되지 않으면 새 로그 블록이 할당가능한지 여부를 확인한다(1408단계). 새 로그 블록으로 할당될 자유 블록이 남아 있으면 그 중 하나를 새 로그 블록으로 할

당하고(1408단계), 자유 블록이 남아있지 않은 경우 블록 병합을 통해 자유 블록을 생성한 다음 생성된 자유 블록을 새 로그 블록으로 할당한다(1409단계). 할당된 로그 블록에 있어서 쓰기가 요청된 페이지와 동일한 오프셋값을 갖는 페이지에 쓰기를 수행한다(1410단계). 다음으로, 로그 포인터 테이블에 대응 엔트리를 생성한다(1405단계).

<93> 도 15는 도 14의 블록 병합을 설명하기 위한 플로우차트이다.

<94> 블록 병합은 앞서 설명한 바와 같이 로그 블록 내의 페이지 배치상태에 따라 각각 다른 방법으로 수행된다. 보다 구체적으로, 프로세서(4)는 로그 블록의 모든 페이지가 대응 데이터 블록의 그것과 동일한 위치에 있는지 여부를 확인하여(1501단계), 그러하면 로그 블록의 모든 페이지가 유효한지 여부를 확인한다(1502단계).

<95> 로그 블록의 모든 페이지가 데이터 블록의 그것과 동일하게 배치되어 있고 모두 유효하면 교환 병합이 수행된다. 한편, 프로세서(4)는 교환 병합을 수행하기 전에 체크 포인트 영역에 복구 정보를 기록한다(1503단계). 다만, 1503단계는 수행되지 않을 수 있다. 수행여부는 시스템 설계자의 선택사항이다. 교환 병합을 위해, 프로세서(4)는 상기 로그 블록을 새로운 데이터 블록이 되도록 맵 영역의 주소변환 정보를 갱신한다(1504단계). 즉, 로그 블록이 새로운 데이터 블록으로 전이되면 사용자 관점에서 논리적 주소에 대응되는 물리적 주소가 변경되므로 주소변환 정보의 갱신이 필요하게 된다. 실제로, 맵 영역의 첫 자유 페이지에 갱신된 주소변환 정보를 기록하면 된다. 맵 영역 또한 마찬가지로 순차적으로 기록되며 더 이상 자유 페이지가 없을 경우에는 자유 블록을 맵 영역으로

할당하여 기록한다. 자유 블록의 할당방법은 도 14를 참조하여 설명한 그것과 동일하게 이루어진다. 이에, 데이터 블록은 지우기-가능 블록으로 전이되며 따라서 데이터 블록을 지운 다음 체크 포인트 영역에 기록된 자유 블록 리스트를 갱신한다(1505단계).

<96> 로그 블록의 어느 한 페이지라도 데이터 블록의 그것과 동일하게 배치되어 있지 않으면 단순 병합이 수행된다. 마찬가지로, 프로세서(4)는 단순 병합을 수행하기 전에 체크 포인트 영역에 복구 정보를 기록한다(1506단계). 1506단계 또한 시스템 설계자의 선택사항이다. 이어서, 자유 블록을 할당받아 여기에 로그 블록의 유효 페이지를 복사하고(1507단계), 나머지 부분은 데이터 블록의 해당 페이지를 기록한 다음(1508단계), 자유 블록이 새로운 데이터 블록이 되도록 맵 영역의 주소변환 정보를 갱신한다(1509단계). 자유 블록의 할당방법은 도 14를 참조하여 설명한 그것과 동일하게 이루어진다. 한편, 로그 블록과 데이터 블록은 지우기-가능 블록으로 전이되며 따라서 로그 블록과 데이터 블록을 지운 다음 체크 포인트 영역에 기록된 자유 블록 리스트를 갱신한다(1510단계).

<97> 로그 블록의 모든 페이지가 데이터 블록의 그것과 동일하게 배치되어 있지 만 데이터 블록의 일부 페이지가 로그 블록에 존재하지 않으면 복사 병합이 수행된다. 마찬가지로, 프로세서(4)는 복사 병합을 수행하기 전에 체크 포인트 영역에 복구 정보를 기록한다(1511단계). 1511단계 또한 시스템 설계자의 선택사항이다. 이어서, 데이터 블록의 유효 페이지를 읽어 로그 블록에 복사한다(1512단계). 다음으로, 로그 블록을 새로운 데이터 블록이 되도록 맵 영역의 주소변환



정보를 갱신하고(1504단계), 데이터 블록을 지운 다음 체크 포인트 영역에 기록된 자유 블록 리스트를 갱신한다(1505단계).

<98> 이처럼, 데이터 갱신을 위한 로그 블록이 발견되지 않은 경우에는 자유 블록을 할당받아 로그 블록으로 전이시키고 여기에 쓰기를 수행한다. 자유 블록이 하나밖에 남지 않아 로그 블록을 할당할 수 없을 경우에는 기존의 로그 블록 중 임의의 하나를 선택하여 블록 병합을 수행함으로써 새로운 자유 블록을 확보하고 난 다음 로그 블록을 할당해야 한다. 이와 같은 상황에서는 블록 병합에 소요되는 비용과 블록들의 향후 사용 가능성을 적절히 고려하여 적절한 선택이 이루어져야 한다. 향후 사용 가능성은 수행하고자 하는 응용 프로그램의 특징에 종속될 수 있다. 본 발명에서는 교체 알고리즘에 대해 특별히 규정하지 않는다. 따라서 본 발명을 구현하기 위해 LRU와 같은 일반화된 교체 알고리즘을 사용하는 것도 가능하다.

#### 【발명의 효과】

<99> 전술한 바와 같이, 본 발명의 목적은 플래시 메모리의 성능을 개선시킬 수 있는 플래시 메모리 관리방법을 제공한다. 종래에는 하나의 데이터 블록의 일부를 갱신하기 위해 나머지 부분까지 복사하거나 대규모의 주소변환 정보를 필요로 하였다. 그러나, 본 발명에 따르면 동일한 페이지에 연속적으로 쓰기가 요청되는 경우에도 하나의 로그 블록 내에서 처리할 수 있기 때문에 플래시 메모리 자원의 효율성을 제고할 수 있다. 나아가, 블록 병합 수행 중에 전원 차단 등을 이유로 시스템이 중단되는 경우에도 일관성있는 데이터의 복원이 가능하다.

**【특허청구범위】**

**【청구항 1】**

플래시 메모리에 소정 데이터를 쓰는 방법에 있어서,

(a) 쓰기가 수행되어 소정 데이터가 기록되어 있는 페이지에 쓰기를 요청 받는 단계;

(b) 상기 페이지가 포함된 데이터 블록에 대응되도록 마련된 로그 블록에 쓰기를 수행하는 단계;

(c) 상기 페이지에 쓰기를 다시 요청받는 단계; 및

(d) 상기 로그 블록 내의 비어있는 자유 페이지에 쓰기를 수행하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 2】**

제1항에 있어서,

상기 (b)단계는

(b11) 비어있는 자유 페이지에 쓰기를 수행하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 3】**

제1항에 있어서,

상기 (b)단계는

(b21) 상기 로그 블록을 할당하는 단계; 및

(b22) 상기 쓰기가 요청된 페이지의 상기 데이터 블록에서의 위치와 동일한 위치의 자유 페이지에 쓰기를 수행하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 4】**

플래시 메모리에 소정 데이터를 쓰는 방법에 있어서,

- (a) 소정 페이지에 쓰기를 요청받는 단계;
- (b) 상기 페이지가 포함된 제1 데이터 블록에 대응되는 제1-1 로그 블록을 할당하는 단계;
- (c) 상기 제1-1 로그 블록 내의 비어있는 자유 페이지에 쓰기를 수행하는 단계;
- (d) 상기 페이지에 쓰기를 다시 요청받는 단계; 및
- (e) 상기 제1-1 로그 블록 내의 비어있는 자유 페이지에 쓰기를 수행하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 5】**

제4항에 있어서,

상기 (b)단계는

- (b1) 제2 데이터 블록 및 이에 대응되는 제2 로그 블록을 기초로 제3 데이터 블록을 생성하는 블록 병합을 수행하는 단계; 및

(b2) 상기 제2 데이터 블록에 대한 지우기를 수행하여 얻어진 자유 블록을 상기 제1 로그 블록으로 할당하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 6】**

제5항에 있어서,

상기 (b1)단계는 상기 제1-1 로그 블록을 할당하기 위한 자유 블록이 존재하지 않을 때 수행되는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 7】**

제5항에 있어서,

상기 (b1)단계는 상기 제1 데이터 블록에 대응되는 기존의 로그 블록이 모두 사용 중일 경우 수행되는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 8】**

제5항에 있어서,

상기 (b1)단계는

(b11) 상기 제2 로그 블록의 페이지와 상기 제2 데이터 블록의 페이지의 배열순서가 동일하고 일대일 대응되는 경우 상기 제2 로그 블록을 상기 제3 데이터 블록으로 전이시키는 교환 병합을 수행하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 9】**

제5항에 있어서,

상기 (b1)단계는

(b12) 상기 제2 로그 블록에 존재하는 페이지들이 모두 한번씩만 쓰기 요청된 경우 상기 제2 로그 블록의 자유 페이지에 상기 제2 데이터 블록의 해당 페이지를 복사하여 상기 제3 데이터 블록을 생성하는 복사 병합을 수행하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

【청구항 10】

제5항에 있어서,

상기 (b1)단계는

(b13) 데이터가 기록되어 있지 않은 자유 블록에 상기 제2 로그 블록에 존재하는 최신 페이지들을 복사하고, 나머지 자유 페이지에 상기 제2 데이터 블록의 해당 페이지를 복사하여 상기 제3 데이터 블록을 생성하는 단순 병합을 수행하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

【청구항 11】

제4항에 있어서,

상기 (e)단계는

(e1) 상기 제1-1 로그 블록 내에 자유 페이지가 존재하지 않을 경우 새로운 제1-2 로그 블록을 할당하는 단계; 및

(e2) 상기 제1-2 로그 블록 내의 자유 페이지에 쓰기를 수행하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 12】**

제11항에 있어서

상기 (e1)단계는

(e11) 상기 제1-1 로그 블록의 페이지와 상기 제1 데이터 블록의 페이지의 배열순서가 동일하고 일대일 대응되는 경우 상기 제1-1 로그 블록을 제2 데이터 블록으로 전이시키는 교환 병합을 수행하는 단계; 및

(e12) 상기 제1 데이터 블록에 대한 지우기를 수행하여 얻어진 자유 블록을 상기 제1-2 로그 블록으로 할당하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 13】**

제11항에 있어서,

상기 (e1)단계는

(e21) 상기 제1-1 로그 블록에 존재하는 페이지들이 모두 한번씩만 쓰기 요청된 경우 상기 제1-1 로그 블록의 자유 페이지에 상기 제1 데이터 블록의 해당 페이지를 복사하여 제2 데이터 블록을 생성하는 복사 병합을 수행하는 단계; 및

(e22) 상기 제1 데이터 블록에 대한 지우기를 수행하여 얻어진 자유 블록을 상기 제1-2 로그 블록으로 할당하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 14】**

제11항에 있어서,

상기 (e1)단계는

(e31) 자유 블록에 상기 제1-1 로그 블록에 존재하는 최신 페이지들을 복사하고, 나머지 자유 페이지에 상기 제1 데이터 블록의 해당 페이지를 복사하여 제2 데이터 블록을 생성하는 단순 병합을 수행하는 단계; 및

(e32) 상기 제1 데이터 블록 또는 상기 제1-1 로그 블록에 대한 지우기를 수행하여 얻어진 자유 블록을 상기 제1-2 로그 블록으로 할당하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 15】**

제11항에 있어서,

상기 (e2)단계는

(e21) 쓰기가 요청된 페이지의 상기 데이터 블록에서의 위치와 동일한 위치의 자유 페이지에 쓰기를 수행하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 쓰기방법.

**【청구항 16】**

플래시 메모리로부터 소정 데이터를 읽는 방법에 있어서,

(a) 로그 포인터 테이블에 요청된 페이지의 논리적 주소 중 블록 주소 부분이 기록된 엔트리를 검색하는 단계;

(b) 검색된 엔트리에 상기 요청된 페이지의 논리적 주소가 기록되어 있는지를 여부를 확인하는 단계; 및

(c) 검색된 엔트리에 기록된 대응 로그 블록의 물리적 주소와, 검색된 논리적 주소의 상기 검색된 엔트리에서의 기록 위치를 참조하여 상기 로그 블록의 해당 페이지에 접근하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 읽기 방법.

【청구항 17】

제16항에 있어서,

상기 (c)단계는

상기 로그 블록의 해당 페이지에 접근함에 있어 상기 검색된 논리적 주소의 상기 검색된 엔트리에서의 기록 위치와 동일한 위치의 페이지에 접근하는 것을 특징으로 하는 플래시 메모리 읽기방법.

【청구항 18】

데이터 블록, 및 상기 데이터 블록을 갱신하기 위한 데이터를 쓰기 위한 로그 블록을 포함하는 플래시 메모리를 관리하는 방법에 있어서,

(a) 제1 데이터 블록의 페이지와 상기 제1 데이터 블록에 대응되는 제1 로그 블록의 페이지의 배열순서가 동일하고 일대일 대응되는 경우 상기 제1 로그 블록을 제2 데이터 블록으로 전이시키는 단계; 및

(b) 주소변환 정보를 갱신하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 관리방법.



**【청구항 19】**

데이터 블록, 및 상기 데이터 블록을 갱신하기 위한 데이터를 쓰기 위한 로그 블록을 포함하는 플래시 메모리를 관리하는 방법에 있어서,

(a) 제1 로그 블록에 존재하는 페이지들이 모두 한번씩만 쓰기 요청된 경우 상기 제1 로그 블록의 자유 페이지에 대응 제1 데이터 블록의 해당 페이지를 복사하여 제2 데이터 블록을 생성하는 단계; 및

(b) 주소변환 정보를 갱신하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 관리방법.

**【청구항 20】**

데이터 블록, 및 상기 데이터 블록을 갱신하기 위한 데이터를 쓰기 위한 로그 블록을 포함하는 플래시 메모리를 관리하는 방법에 있어서,

(a) 데이터가 기록되어 있지 않은 자유 블록에 제1 로그 블록에 존재하는 최신 페이지들을 복사하고, 나머지 자유 페이지에 대응 제1 데이터 블록의 해당 페이지를 복사하여 제2 데이터 블록을 생성하는 단계; 및

(b) 주소변환 정보를 갱신하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 관리방법.

**【청구항 21】**

제18항 내지 제20항 중 어느 한 항에 있어서,

상기 (a)단계 이전에

(a0) 상기 (a)단계 또는 (b)단계의 수행 중 시스템이 중단되는 경우 데이터를 복구하기 위한 복구 정보를 기록하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리 관리방법.

**【청구항 22】**

제21항에 있어서,

(c) 상기 (a)단계 또는 (b)단계의 수행 중 시스템이 중단되는 경우 상기 복구 정보를 참조하여 데이터를 복원하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리 관리방법.

**【청구항 23】**

제22항에 있어서,

상기 복구 정보는 상기 자유 블록의 리스트, 로그 블록의 리스트, 상기 로그 블록을 관리하기 위한 데이터 구조인 로그 포인터 테이블을 포함하는 것을 특징으로 하는 플래시 메모리 관리방법.

**【청구항 24】**

제23항에 있어서,

상기 로그 포인터 테이블에는 로그 블록에 대응되는 갯수의 로그 포인터 테이블 엔트리가 구성되어 있고, 각 엔트리에는 대응 데이터 블록의 논리적 주소와 해당 로그 블록의 물리적 주소가 매핑되어 있고, 해당 로그 블록 내의 각 페이지들의 물리적 배열 순서에 따라 해당 데이터 블록의 요청된 페이지의 논리적 주소가 기록되어 있는 것을 특징으로 하는 플래시 메모리 관리방법.

**【청구항 25】**

제23항에 있어서,

상기 로그 포인터 테이블은 상기 로그 블록들이 기록된 로그 블록 영역을 스캔하여 필요한 정보를 얻어 구성되는 것을 특징으로 하는 플래시 메모리 관리 방법.

**【청구항 26】**

데이터 블록, 및 상기 데이터 블록을 갱신하기 위한 데이터를 쓰기 위한 로그 블록을 포함하는 플래시 메모리를 관리하는 방법에 있어서,

(a) 플래시 메모리의 소정 영역을 할당하고 할당된 영역에 복구 정보로서 상기 데이터 블록 및 상기 로그 블록의 리스트, 및 상기 로그 블록을 관리하기 위한 데이터 구조를 기록하는 단계;

(b) 시스템이 중단되는 경우 상기 복구 정보를 기초로 현재 플래시 메모리에 기록된 상태를 점검하여 오류 발생 여부를 확인하는 단계; 및

(c) 오류가 발생된 경우 상기 복구 정보를 참조하여 데이터를 복원하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 관리방법.

**【청구항 27】**

제26항에 있어서,

상기 복구 정보는 자유 블록의 리스트를 더 포함하는 것을 특징으로 하는 플래시 메모리 관리방법.

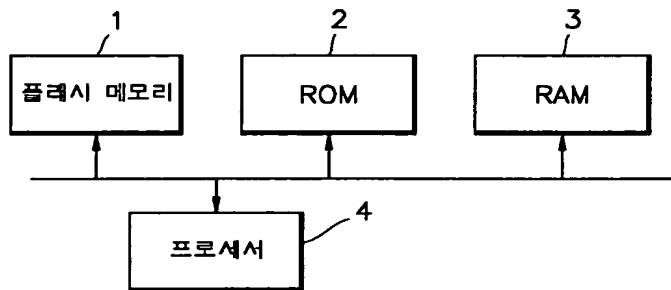
**【청구항 28】**

제27항에 있어서,

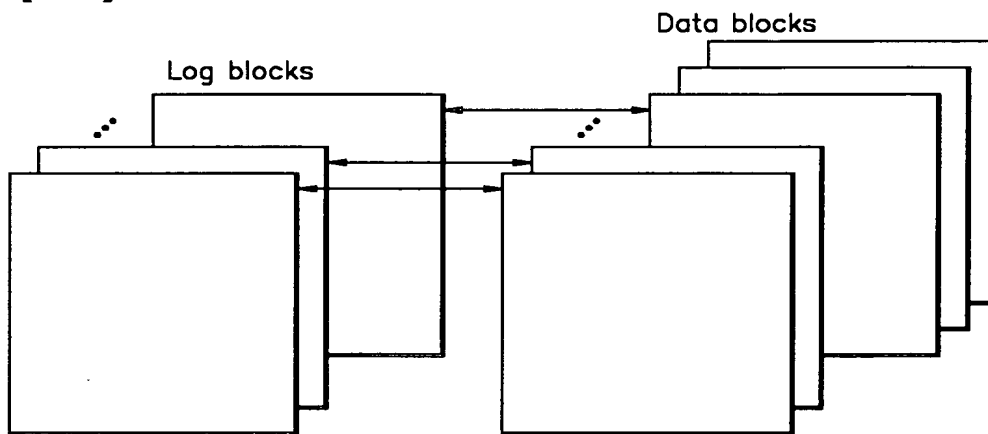
상기 로그 포인터 테이블에는 로그 블록에 대응되는 갯수의 로그 포인터 테이블 엔트리가 구성되어 있고, 각 엔트리에는 대응 데이터 블록의 논리적 주소와 해당 로그 블록의 물리적 주소가 매핑되어 있고, 해당 로그 블록 내의 각 페이지들의 물리적 배열 순서에 따라 해당 데이터 블록의 요청된 페이지의 논리적 주소가 기록되어 있는 것을 특징으로 하는 플래시 메모리 관리방법.

【도면】

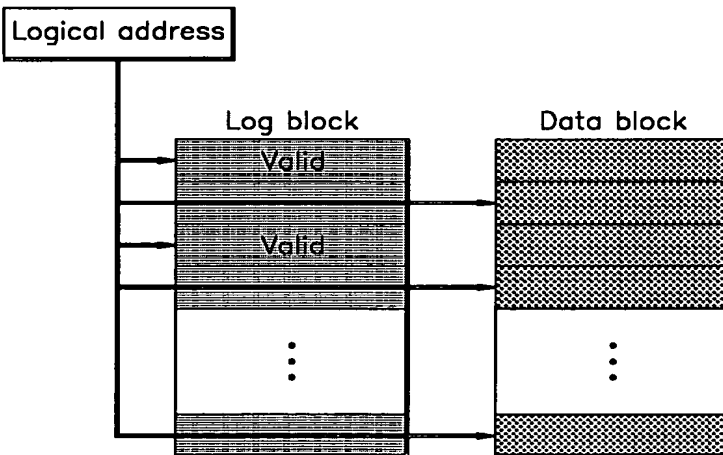
【도 1】



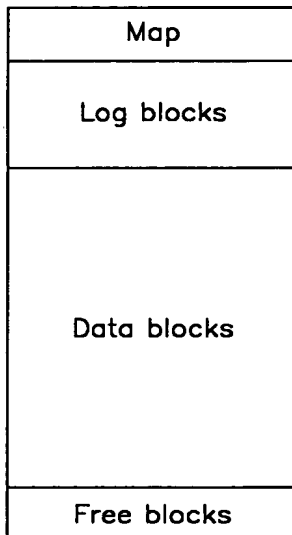
【도 2】



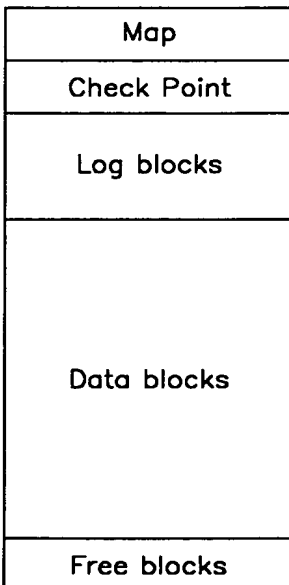
【도 3】



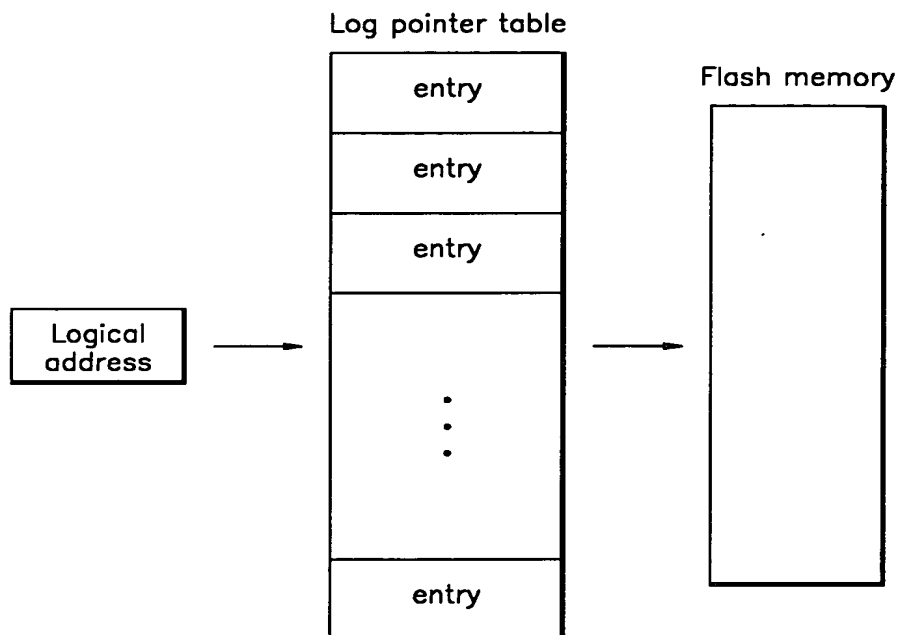
【도 4】  
Flash memory



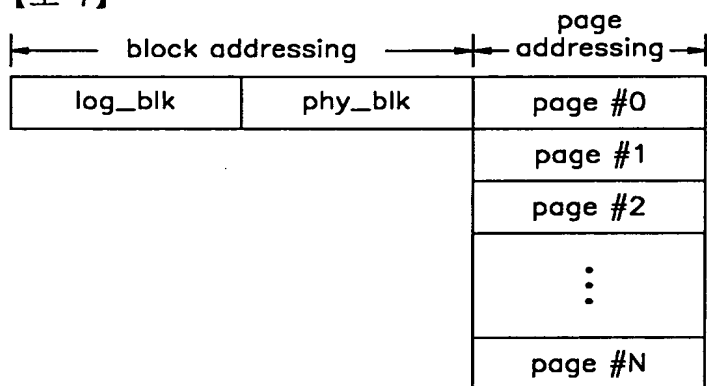
【도 5】  
Flash memory



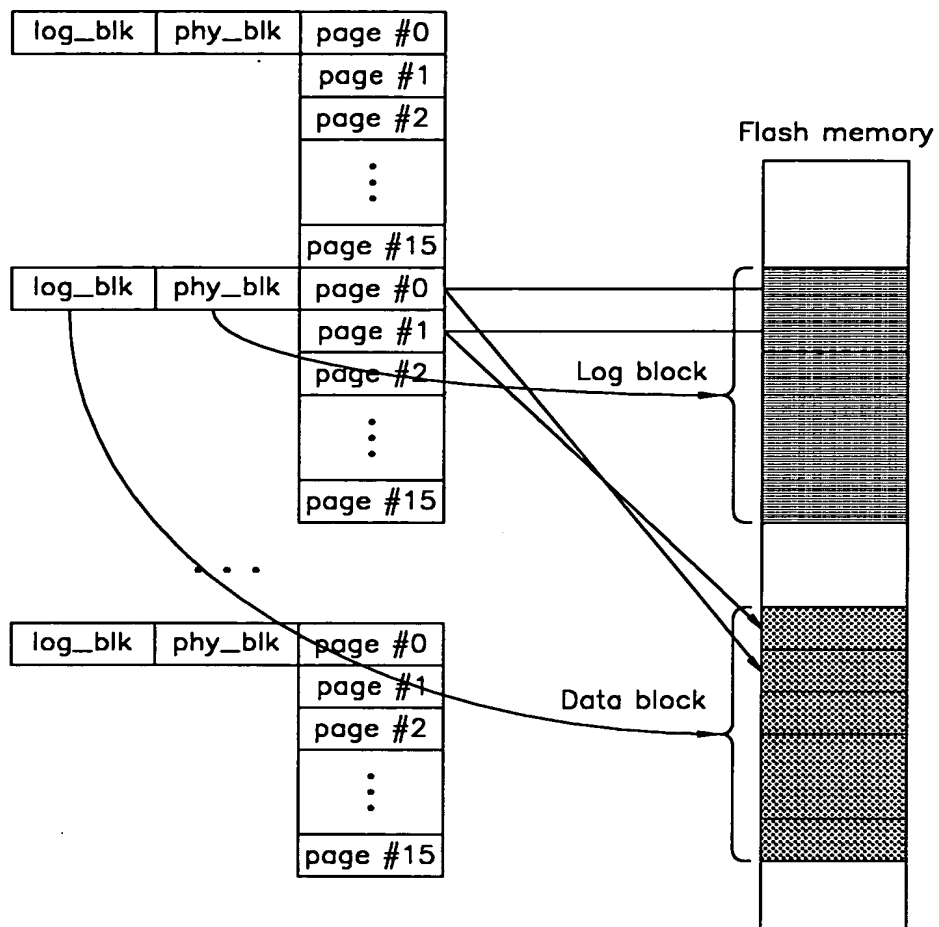
【도 6】



【도 7】

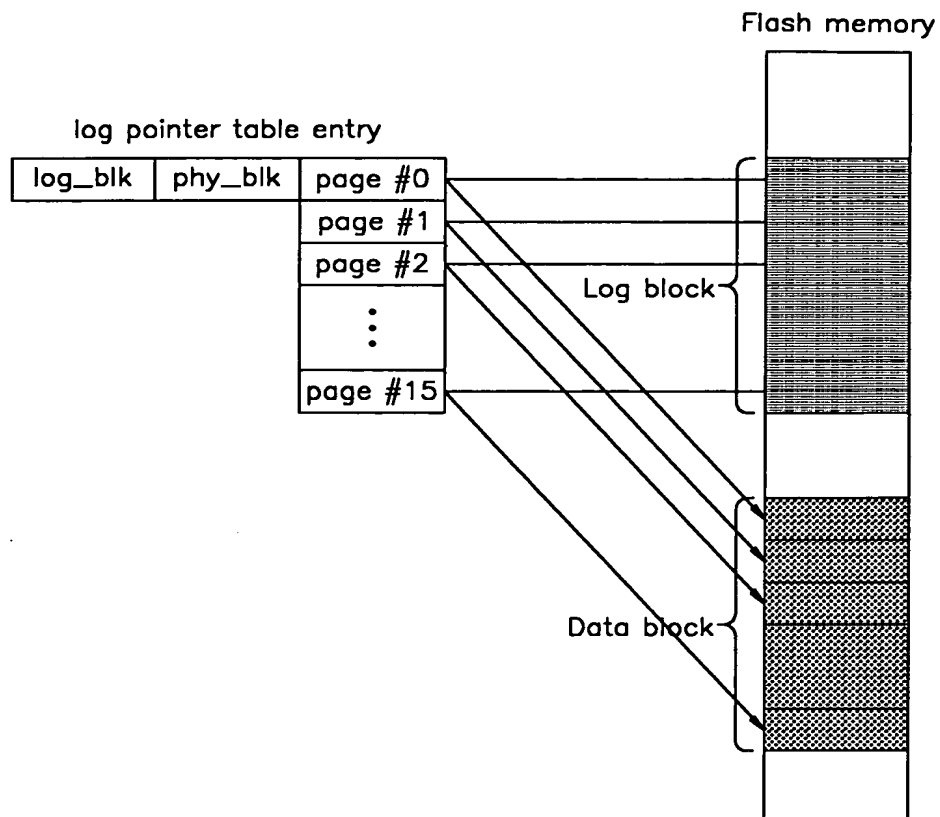


【도 8】

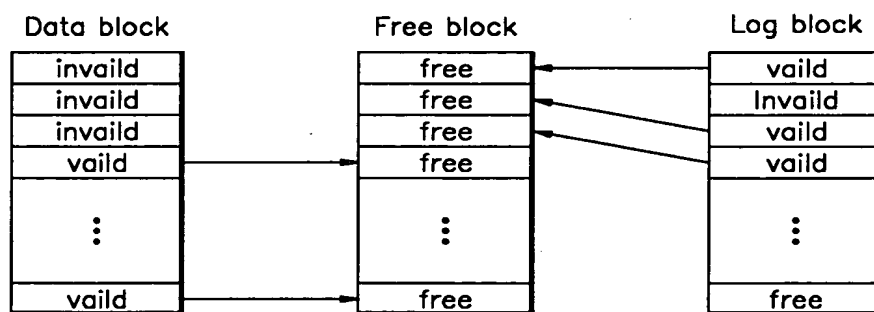




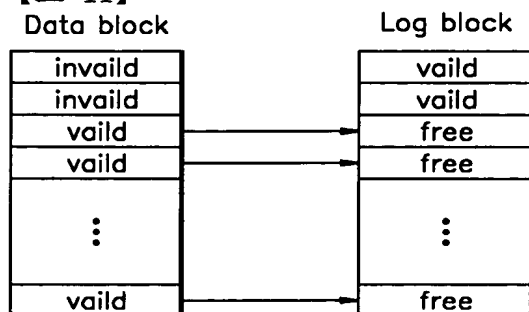
【도 9】



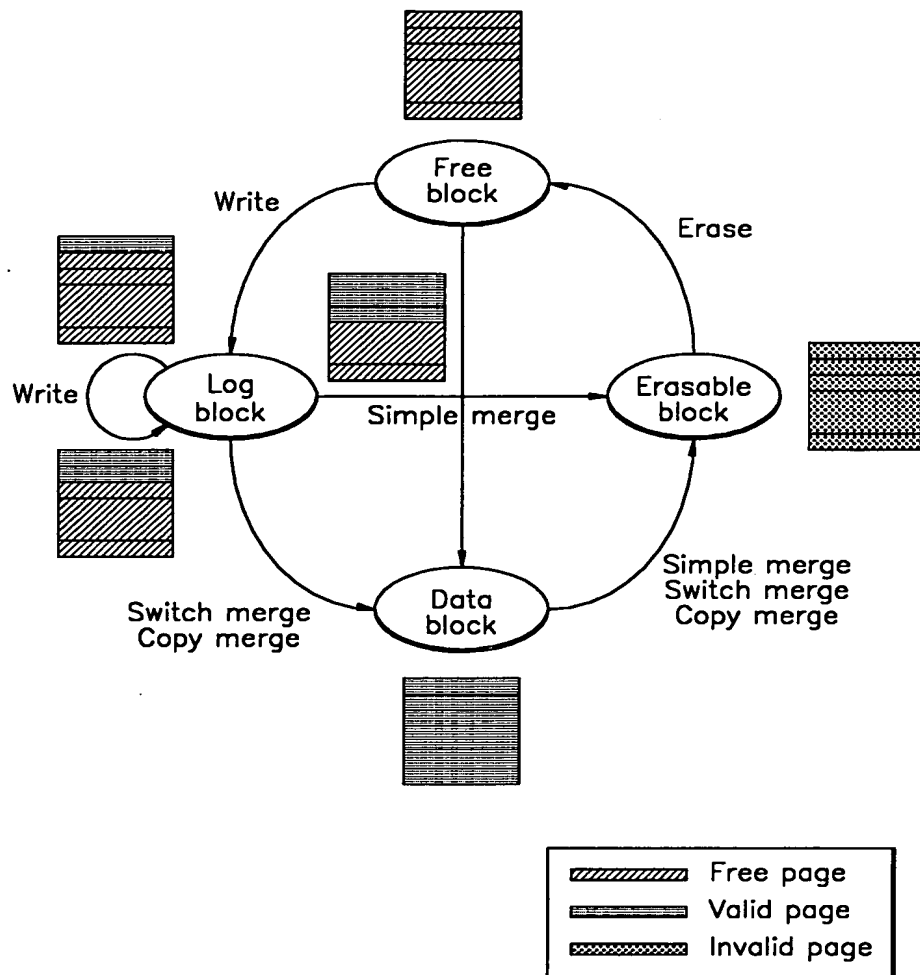
【도 10】



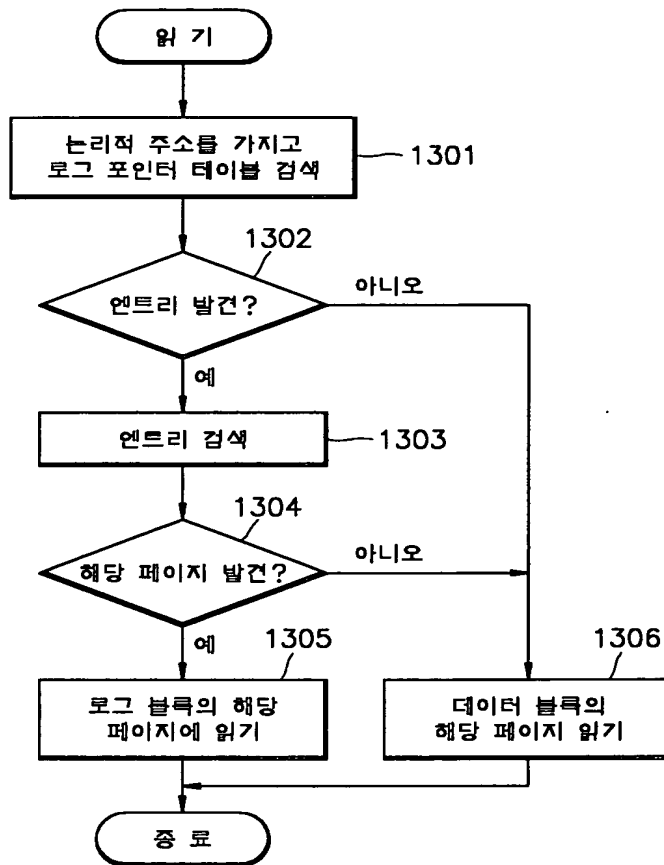
【도 11】



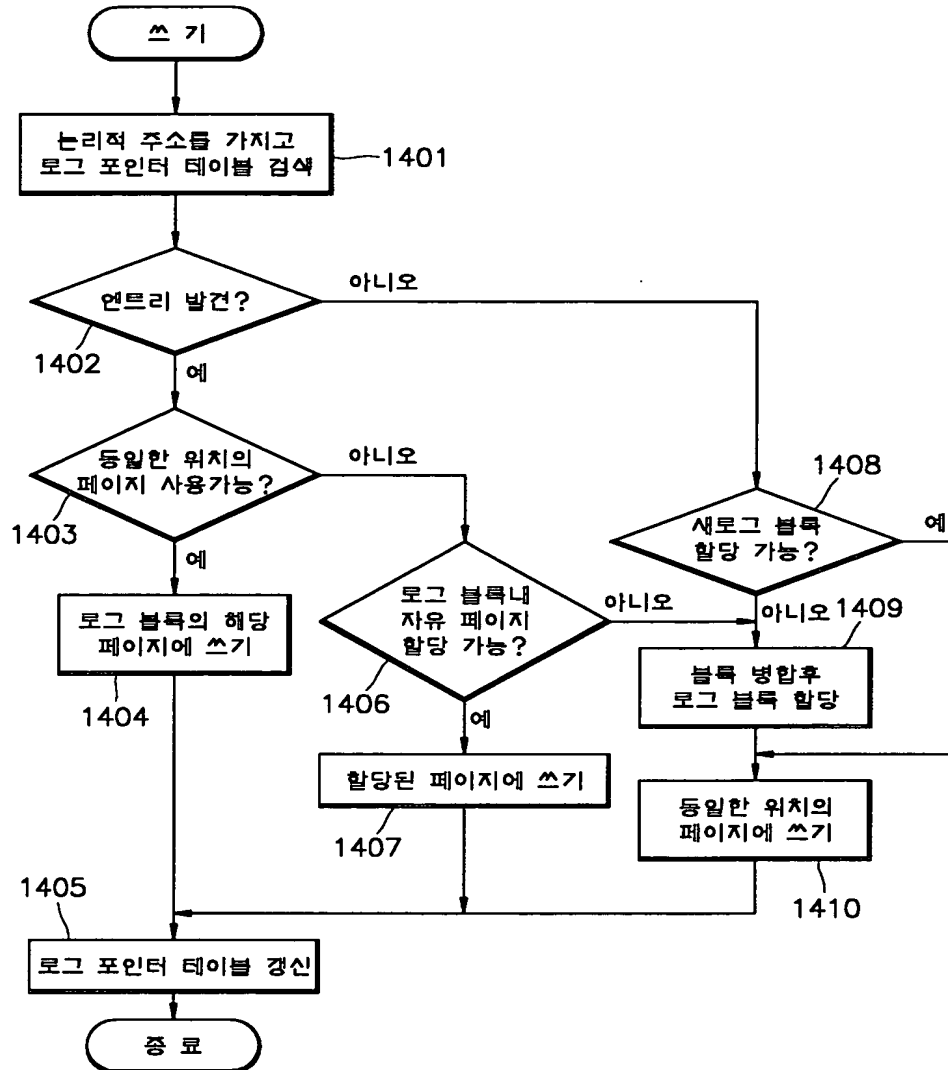
【도 12】



【도 13】



【도 14】



【도 15】

